| **Feature** | **Virtual DOM** | **Real DOM** |
| --- | --- | --- |
| Definition | An abstract representation of the browser's Document Object Model (DOM) | The actual Document Object Model (DOM) as it exists in the browser |
| Update process | Only updates the parts of the DOM that have changed | Updates the entire DOM, even if only a small part has changed |
| Performance | Typically faster than the Real DOM, as it reduces the number of updates needed | Can be slower than the Virtual DOM, as it may require frequent updates to the entire DOM |
| Memory usage | Typically uses more memory than the Real DOM, as it stores a copy of the DOM in memory | Typically uses less memory than the Virtual DOM, as it only stores the actual DOM in memory |
| Initial load time | Slightly slower than the Real DOM, as it requires an initial render to create the virtual representation | Slightly faster than the Virtual DOM, as it does not require an initial render to create the actual DOM |
| Accessibility | May require additional work to ensure accessibility standards are met, as it does not generate semantic HTML directly | Generates semantic HTML directly, making it easier to ensure accessibility standards are met |
| Preferred use case | Best suited for large and complex applications with frequent updates and dynamic content | Best suited for small and simple applications with infrequent updates and static content |

DOM in simpler terms.

The Document Object Model, or DOM, is the way that web browsers represent and manipulate web pages. The DOM is a tree-like structure that represents the HTML elements and their relationships to one another.

The Real DOM is the actual tree-like structure that represents the HTML elements in the web browser. When changes are made to the page, the Real DOM is updated to reflect those changes. However, updating the Real DOM can be slow and memory-intensive, especially for large and complex web applications.

The Virtual DOM is an abstract representation of the Real DOM that is stored in memory. When changes are made to the page, they are first made to the Virtual DOM, which is then compared to the Real DOM to determine what changes need to be made. This process is faster and more memory-efficient than updating the Real DOM directly.

Here's an example: Let's say you have a web page with a button that changes color when clicked. With the Real DOM, every time the button is clicked, the entire DOM needs to be updated to reflect the new color. With the Virtual DOM, only the part of the tree that represents the button needs to be updated, which is much faster and more efficient.

In general, the Virtual DOM is better suited for large and complex web applications that require frequent updates, while the Real DOM is better suited for simple applications with infrequent updates.